Laboratório de Circuitos Lógicos

Rafael Rocha Matias
José Maria Pires de Menezes Junior

04 de setembro de 2011

SUMÁRIO

Capítulo 1—Como Realizar a Preparação do Relatório 4

1.1 Formatação do Texto 4
1.2 Conteúdo da Preparação 4
1.3 Montagens 5

Capítulo 2—Guia do Experimento 1: Portas Lógicas 6

2.1 Objetivos 6
2.2 Introdução 6
2.2.1 Funções Lógicas 7
2.3 Verificando o funcionamento de um circuito lógico 14
2.4 Cuidados na montagem e desmontagem do circuito 15
2.5 Depuração do Circuito 15
2.6 Montagens 16
2.6.1 1ª Montagem: Porta NOR de duas entradas 16
2.6.2 2ª Montagem: Porta AND de três entradas 16
2.6.3 3ª Montagem: Porta XOR/XNOR 16
2.7 Questões 16
2.8 Referências Bibliográficas 17

Capítulo 3—Relatório do Experimento 1: Portas Lógicas 18

3.1 Material Utilizado 18
3.2 Resumo da teoria 18
3.3 1ª Montagem 18
3.3.1 Descrição do Funcionamento 18
3.3.2 Diagrama elétrico 19
3.3.3 Verificação do Funcionamento 19
3.4 2ª Montagem: Porta AND de três entradas 20

3.4.1 Descrição do Funcionamento 20
3.4.2 Diagrama elétrico 20
3.4.3 Verificação do funcionamento 21
3.4.4 Diagrama Elétrico 21
3.4.5 Verificação do Funcionamento 21
3.5 Conclusões 21
3.6 Questões 21

Capítulo 4—2º Experimento: Equivalências de portas lógicas 22

4.1 Objetivos 22
4.2 Introdução 22
4.3 Preparação do Relatório 25
4.3.1 1ª Montagem: Porta XOR a partir de Porta NAND. 25
4.3.2 2ª Montagem:Comparador de Magnitude 25
4.3.3 3ª Montagem: Porta NAND a partir de porta OR 26
4.3.4 Questões 26

Capítulo 5—3º Experimento: Somador Binário 27

5.1 Objetivos 27
5.2 Introdução 27
5.3 Preparação do Relatório 28
5.4 4 Montagens 29
5.4.1 1ª Montagem: Seletor de Função Igualdade/Complemento de 1. . 29
5.5 2ª Montagem: Soma de Números sem Sinal. 30
5.5.1 3ª Montagem: Detetor de Estouro de Capacidade 30
5.5.2 4ª Montagem: Somador - Subtrrator 31
5.6 Questões 32

Capítulo 6—4º Experimento Circuitos Combinacionais 33

6.1 Objetivos: 33
6.2 Prepação do Relatório 33
6.3 Montagens 33
6.3.1 1ª Montagem:Circuito gerador de paridade impar 33
6.3.2 2ª Montagem:Circuito verificador de paridade impar 34
6.3.3 3ª Montagem: Comparador de 3 bits 34

6.3.4 4ª Montagem: Semáforo 35
6.4 Questões 36

COMO REALIZAR A PREPARAÇÃO DO RELATÓRIO

1.1 FORMATAÇÃO DO TEXTO

A formatação da preparação do relatório deve obdecer os seguintes critérios:

- Os textos devem ser apresentados em papel branco, formato A4 (21 cm x 29,7 cm), digitados, datilografados ou manuscritos.
- Utilizar fonte de tamanho 11 ou 12.
- Não utilizar o papel frente e verso.
- Margens superior e esquerda de 3 cm.
- Margens inferior e direita de 2 cm.
- Todas as páginas, a partir da folha de rosto, devem ser contadas seqüencialmente, mas não numeradas.
- A Numeração das páginas deve estar no canto superior direito, estando o último algarismmo a 2 cm da borda direita da folha.
- As citações devem ser apresentadas conforme a ABNT NBR 10520.
- Todo o texto deve ser digitado ou datilografado com espaço 1,5, entrelinhas.

1.2 CONTEÚDO DA PREPARAÇÃO

A preparação de cada relatório deve ser realizada de acordo com o guia de cada experimento e deve ter as seguintes seções e subeseções:

1. **Objetivo(s):** Listar o(s) objetivo(s) do experimento.
2. **Material Utilizado:** Listar os circuitos integrados e demais componentes utilizados, incluir o módulo de treinamento didático.
3. **Resumo da teoria:** Apresentar de modo consiso, um resumo da teoria empregado no experimento.

4. **Montagens:** Para cada montagem solicitada no guia, elaborar os seguintes subseções:

   (a) **Descrição do Funcionamento:** Descrever o funcionamento do circuito a ser montado e/ou seu projeto lógico.

   (b) **Diagrama Elétrico:** Desenhar o diagrama elétrico correspondente.

   (c) **Verificação do funcionamento:** Descrever a verificação do funcionamento do circuito, em geral realizada por meio de tabelas.

5. **Conclusões:** Realizar comentários sobre o experimento.

6. **Questões:** Enunciar e resolver as questões solicitadas no guia.

1.3 MONTAGENS

As subseções de cada montagem devem ser detalhadas da seguinte forma:

1. **Descrição do Funcionamento:** Consiste em descrever o funcionamento do circuito lógico por meio de um *texto sucinto*. Completar o texto com os seguintes ítens:

   - Bloco Funcional/Lógico (ou uma hierarquia de blocos) e/ou Circuito lógico.
   - Tabela(s) da Verdade esperda(s) (completa ou simplificada).
   - Expressões lógicas.
   - No caso de ser realizado o projeto do circuito lógico, todos os passos do projeto devem ser incluídos (Bloco(s), Tabelas, Mapas, Expressões, Circuito Lógico).

2. **Diagrama Elétrico:** Consiste em desenhar o diagrama elétrico correspondente ao circuito lógico usado, indicando os nomes das variáveis, bem como as chaves e os leds usados na montagem.

3. **Verificação do Funcionamento:**

   - Preparar tabelas(s) para verificação prática do funcionamento do circuito. No cabe çalho da tabela, indicar também as chaves e os leds correspondentes a cada entrada e saída.
   - Montar o circuito e verificar o seu funcionamento.

GUIA DO EXPERIMENTO 1: PORTAS LÓGICAS

2.1 OBJETIVOS

- Usar a Lógica e a Álgebra de Boole de 2 valores para modelar sistemas digitais;
- Descrever e implementar as funções lógicas elementares por meior de portas lógicas elementares;
- Construir Tabelas verdade e Tabelas funcionais;
- Construir e utilizar diagramas:
  - Lógicos;
  - De Pinos;
  - Elétrico;
- Montar um circuito lógico, testar o seu funcionamento e desmontá-lo, tomando os cuidados necessários;
- Depurar um circuito lógico que não funcione como esperado;

2.2 INTRODUÇÃO

A informação digital, é normalmente, representada simbolicamente por meio de códigos numéricos binários. Nesses códigos, a unidade da informação é o bit (contração de *binary digit*), que pode assumir o valor **1** ou o valor **0**. O processamento da informação codificada nessa forma é realizada por sistemas digitais (binários) que podem ser descritos por funções binárias de variáveis digitais binárias. Essas variáveis binárias correspondem a cada um dos bits da informação.

Os sistemas digitais podem ser modelados usando a Lógica. Uma variável lógica pode assumir um *valor da verdade* que também corresponde a apenas dois valores: **V** (Verdadeiro) e **F** (Falso). Algumas vezes esses valores são representados, respectivamente, pelos dígitos **1** e **0**.

2.2.1 Funções Lógicas

A NEGAÇÃO é uma função lógica unária (de uma variável), sendo representada por meio do operador unário *barra* ($\overline{\ }$) colocado acima da variáve, correspondendo à seguinte expressão lógica: $Z = f(A) = \overline{A}$.

Existem algumas funções lógicas binárifas (de duas variáveis) que, juntamente com a negação, formam um conjunto de funções lógicas elementares, e a partir das quais qualquer função lógica de n variáveis pode ser obtida.

A função lógica *E (AND)* que corresponde ao conectivo lógica &. Duas declarações relacionadas por & formam uma declaração composta cujo *valor-da-verdade* é **V** se e somente se o *valor-da-verdade* de **todas** as componentes for **V**. A operação lógica *E ou AND* é representada pelo operador binário '$\wedge$' ou '$\cdot$'.

A função OU (OR) corresponde ao conectívo lógico **ou**, se duas declarações estiverem relacionadas por esse conectívo formam uma declaração composta cujo *valor-da-verdade* é **V** se e somente se o *valor-da-verdade* de **pelo menos uma** das componentes for **V**. A operação lógica **OR** ou **OU** é representada pelo operador binário $\vee$ ou $+$.

É possível mostrar que as demais funções lógicas, de duas ou mais variáveis, podem ser obtidas a partir apenas das três funções lógicas elementares: NEGAÇÃO, AND e OR.

A Álgebra de Boole de dois valores é uma ferramenta matemática que permite a manipulação das expressões (booleanas ou lógicas) de uma forma algébrica.

A lógica e a álgebra de Boole de dois valores podem ser consideradas como sistemas matemáticos equivalentes por meio da seguinte analogia:

$$\begin{aligned}
\text{Valor VERDADEIRO} &\iff \text{valor 1} \\
\text{valor FALSO} &\iff \text{valor 0} \\
\text{Operação AND } (\wedge) &\iff \text{operação produto Booleano } (\cdot) \\
\text{Operação OR } (\vee) &\iff \text{operação soma Booleana } (+) \\
\text{Operação NEGAÇÃO } (\overline{\ }) &\iff \text{operação complemento } (\overline{\ })
\end{aligned}$$

A avaliação de uma função lógica consiste em indicar o *valor-da-verdade* da funlçai oara cada possível combinação de *valores-da-verdade* das variáveis. COmo essas variáveis são discretas, podendo assumir apenas dois valores, o número de possíveis combinações é finito e igual a uma potência de dois, ou seja, é igual a 2^n, onde n é o número de variáveis.

Assim, uma função lógica também pode ser representada por uma tabela denominada Tabela da Verdade que possui uma linha para cada combinação de valores das variáveis. Na Tabela 2.1 são representadas as Tabelas da Verdade das funções NEGAÇÃO, AND e OR. Observar que, segundo a lógica, essas tabelas são preenchidas com os valores Ve

F. Porém, também é comum usar osvalores simbólicos 1 e 0, usando a analogia com a aÁlgebra de Boole de dois valores.

(a) NEGAÇÃO

A	$\overline{A}$	A	$\overline{A}$
F	V	0	1
V	F	1	0

(b) AND

A	B	A ∧ B	A	B	A·B
F	F	F	0	0	0
F	V	F	0	1	0
V	F	F	1	0	0
V	V	V	1	1	1

(c) OR

A	B	A ∨ B	A	B	A+B
F	F	F	0	0	0
F	V	V	0	1	1
V	F	V	1	0	1
V	V	V	1	1	1

Tabela 2.1 Tabelas da Verdade para as funções NEGAÇÃO, AND e OR.

Aplicando a negação ao AND obtém-se a função NAND (NOT AND), que corresponde à seguinte expressão lógica: $Z = f(A, B) = \overline{A \wedge B}$. Analogamente, aplicando a negação ao OR obtém-se a função NOR (NOT OR), que corresponde à seguinte expressão lógica: $Z = f(A, B) = \overline{A \vee B}$. Na Tabela 2.2 são apresentadas as Tabelas da Verdade para o NAND e o OR.

(a) NAND

A	B	$\overline{A \wedge B}$	A	B	$\overline{A \cdot B}$
F	F	V	0	0	1
F	V	V	0	1	1
V	F	V	1	0	1
V	V	F	1	1	0

(b) NOR

A	B	$\overline{A \vee B}$	A	B	$\overline{A + B}$
F	F	V	0	0	1
F	V	F	0	1	0
V	F	F	1	0	0
V	V	F	1	1	0

Tabela 2.2 Tabelas da Verdade para as funções NAND e NOR.

Além dessas funções, há duas outras que, sob certas condições, também podem ser consideradas elementares. A primeira é a função OU-EXCLUSIVO (XOR) que corresponde à interpretação do conectivo **ou** associado à exclusão mútua, ou seja, as duas condições não podem ser verdadeiras simultaneamente. Duas declarações relacionadas por *ou exclusivo* formam uma declaração composta cujo *valor-da-verdade* é V se e somente se o *valor-da-verdade* de apenas uma das componentes for V. O XOR é representada por $Z = f(A, B) = A \oplus B$.

Aplicando a NEGAÇÃO a função XOR, obtem-se a função XNOR (NOT XOR). Essa função também é denominada de EQUIVALÊNCIA ou COINCIDÊNCIA, pois o *valor-da-verdade* da função é V se e somente se os *valores-da-verdade* das componentes forem iguais, ou seja, ambos V ou ambos F. O XNOR é uma função lógica binária fque corresponde à seguinte expressão lógica: $Z = f(A, B) = \overline{A \oplus B}$. A interpretação como

Equivalência ou Coincidência é representada pelo operador binário $\odot$. Na Tabela 2.3 é apresentada as Tabelas da Verdade das funções XOR e XNOR.

(a) XOR

A	B	A $\oplus B$	A	B	A $\oplus$ B
F	F	F	0	0	0
F	V	V	0	1	1
V	F	V	1	0	1
V	V	F	1	1	0

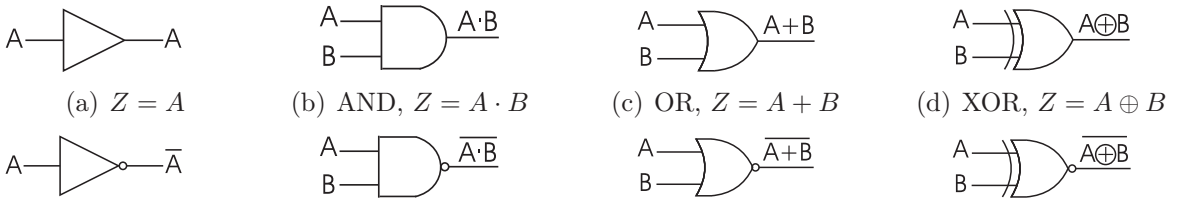
(b) XNOR

A	B	$\overline{A \oplus B}$	A	B	A$\odot$ B
F	F	V	0	0	1
F	V	F	0	1	0
V	F	F	1	0	0
V	V	V	1	1	1

Tabela 2.3 Tabelas da Verdade para as funções NAND e NOR.

Uma função lógica também pode ser representada graficamente por um diagrama de blocos denominado de Diagrama Lógica ou Circuito Lógico. O diagrama lógico pode consistir em um único bloco funcional que simbolize a relação entre as variáveis de entrada e a variável de saída (valor da função), ou, quando a função lógica é descrita a partir de uma associação de funções elementares, seu diagrama lógico pode mostrar como os blocos lógicos elementares devem ser interligados de modo a produzir a resposta desejada.

Na Figura 2.1 é apresentado os blocos funcionais das funções lógicas.



(a) $Z = A$ (b) AND, $Z = A \cdot B$ (c) OR, $Z = A + B$ (d) XOR, $Z = A \oplus B$

(e) $Z = \overline{A}$ (f) NAND, $Z = \overline{A \cdot B}$ (g) NOR, $Z = \overline{A + B}$ (h) XNOR, $Z = \overline{A \oplus B}$

Figura 2.1 Funções Lógicas (Bloco lógico e expressão lógica).

Qualquer função lógica de **n** variáveis pode ser implementada na forma de um dispositívo eletrônico, no entanto, um número limitado de funções é fornecido pelos fabricantes. As demais funções devem ser implementadas atr4avés da combinação das funções básicas, ou seja, funções mais complexas são implementadas a partir das funções mais simples.

Os dispositivos eletrônicos que implementam a funções lógicas elementares são denominados de Portas Lógicas. O nome *porta* advém do processamento da informação realizado por um circuito lógico ser interpretado com um fluxo de dados que progride a partir das entradas do sistema, passando pelos circuito intermediários, até produzir uma resposta na saída. A porta lógica é usada, então, para controlar a passagem ou não desse fluxo de dados.

Vale lembrar que os dispositívos eletrônicos necessitam de serem alimentados a partir de uma fonte de tensão contínua externa e, portanto, possuem dois outros terminais de entrada para essa alimentação — VCC e GND (ground⇒ Terra) — que, geralmente, não aparecem no diagrama lógico.

Para construir um dispositívo eletrônico que implemente uma função lógica é necessário representar as variáveis lógicas/binárias por meio de grandezas físicas elétricas como a tensão, a corrente, a carga elétrica ou o sentido de magnetização de uma terial magnético. o mais comum é usar a tensõ elétrica presente no terminal de entrada ou de saída do dispositivo para representar essas variáveis. Essa variável física também é discreta, podendo assumir dois valores: um nível alto, representado por **H** (High), e um nível baixo, representado por **L** (Low).

A descrição do funcionamento de uma porta lógica é realizada por meio da Tabela de Funcionamento, que é semelhante à Tabela da Verdade, sendo que os valores usados para as entradas e saídas são os níveis **H** e **L**. Por exemplo, a porta AND é um dispositivo que funciona de acordo com a tabela de funcionamento dada na Tabela 2.4. A interpretação que associa o nível ao valor 1 e o nível L ao valor 0 é denominada de Lógica Positiva, uma vez que preserva a hieraquia, ou seja, H está acima de L, bem como 1 está acima de 0.

Aplicando essa lógica, a porta descrita pela Tabela 2.4(a) realmente é uma porta AND, como mostra a Tabela 2.4(b). Na maioria das aplicações, é a lógica positiva que é usada. Entretanto, se for usada a Lógica Negativa, que associa o nível H ao valor 0 e o nível L ao valor 1, esse dispositivo corresponderá a uma porta OR, como mostrado na Tabela 2.4(c).

(a) Dispositivo

A	B	Z
L	L	L
L	H	L
H	L	L
H	H	H

(b) AND-Logica Positiva

A	B	Z
0	0	0
0	1	0
1	0	0
1	1	1

(c) OR-Lógica Negativa

A	B	Z
1	1	1
1	0	1
0	1	1
0	0	0

Tabela 2.4 Porta Lógica: (a) funcional, (b) Lógica Positiva, (c) Lógica Negativa.

As portas lógicas são, normalmente, encontradas com duas, três, e quatro entradas (há ainda algumas portas com cico, oito, doze e treze entradas), com exceção das portas XOR e XNOR que é encontrada com apenas duas entradas (há portas XOR de oito e de nove entradas para aplicações geração/detecção de paridade par/impar).

Quando for necessária uma função lógica com um determinado número de variáveis de entradas, e não houver uma porta lógica com o mesmo número de entradas, há dois caminhos a seguir:

a) Se o número de entradas da porta for maior que o número de variáveis, deve-se conectar as entradas excedentes ao GND ou ao VCC, tal que essa(s) entrada(s) não altere(m) o cálculo da expressão lógica.

b) Se o número de entradas da porta for menor que o número de variáveis, deve-se fazer a associatividade (apenas algumas funções lógicas possuem essa propriedade) e interligar portas com um número de entradas de menor, de modo a obter o número desejado.

Por exemplo, a função AND de três entradas, indicada na Figura 2.2(a), pode ser obtida a partir de duas portas AND de duas entradas, como indicado na Figura 2.2(b), pois pela lei da associatividade, $Z = A \cdot B \cdot C = (A \cdot B) \cdot C$. Usando uma porta AND de quatro entradas, também obtém-se essa função, bastando ligar uma das entradas em VCC, como indicado na Figura 2.2(c).

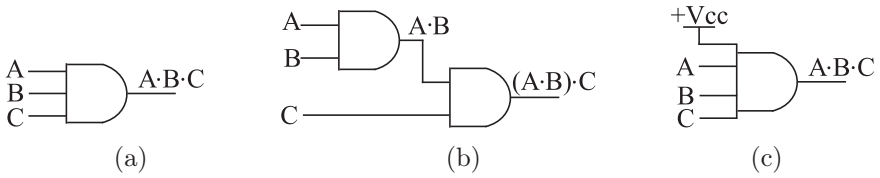


Figura 2.2 Função Lógica AND de três entradas.

As portas lógicas são implementadas utilizando-se dispositivos semicondutores tais como transistores e diodos. Várias portas lógicas de um mesmo tipo são implementadas em um bloco monolítico de material semicondutor (chip). Em seguida, esse bloco é interligado a terminais denominados de pinos e o conjunto encapsulado com plástico ou cerâmica. Esse processo é chamado de integração e o dispositivo produzido é um circuito integrado digital. A quantidade de portas lógicas em cada circuito integtrado (CI) depende do número de variáveis da função e do númerod de pounos do CI.

A identificação do circuito é realizada por meio de um código escrito na parte de cima do encapsulamento e da identificação da função de cada pino é dada por meio de diagrama de pinos (pinagem). Os fabricantes fornecem um manual que deve sempre ser consultado para obter informações sobre as características elétricas e sobre a pinagem de cada CI.

Os CI's construídos com tecnologia TTL (Lógica a transistor-Transistor) possui duas séries de circuitos e é identificada pelos dois primeiros dígitos do código do CI: 74XX

significa que o dispositivo tem especificações comerciais e uma faixa de temperatura de operação de 0 a 70 ºC, enquanto que 54XX significa que as especificações são militares e que pode operar entre -55 a 125 ºC. Os CI's dessas duas séries possuem, normalmente, o mesmo diagrama de pinos.

A família TTL possui diversas subfamílias, cada uma com características próprias em termos de níveis de tensão, margem de ruído, correntes, fan-out, dissipação de energia, velocidade (tempos de atraso), etc. A identificação da subfamilia é feita por meio de letras que seguem os dígitos 54 ou 74, por exemplo:

Sem letras	TTL Padrão
LS	Schottky de baixa potência
L	Baixa potência
S	Schottky
H	Alta potência
ALS	Schottky de baixa potência avançada
AS	Schottky avançada
F	Schottky avançada fairchild (FAST)

Após as letras de identificação da subfamília, normalmente estão presentes dois ou três dígitos (eventualmente seguidos por uma letra A, B, etc) que identificam a função lógica implementada. Por fim, podem vir uma ou duas letras que identificam o tipo de encapsulamento. Observar que a convenção de identificação dos CI's varia com o fabricante.

Os circuitos integrados utilizados no laboratório são construídos com um encapsulamento no formato DIL (Dual-in-Line) que possui duas carreiras paralelas de pinos. Os pinos são numerados, sendo que o pino 1 é identificado por uma marca ou por uma ranhura na lateral esquerda do CI, como mostra o CI da Figura 2.3(a). Os demais pinos da linha inferior são numerados da esquerda para a direita, enquanto que os pinos da linha superior são numerados da direita para a esquerda, comindacado no diagrama de pinos d Figura 2.3(b) para um CI de 14 pinos.

Para os CI's que implementam portas lógicas, é comum desenhas as portas no diagram de pinos, mostrando claramente a disposição física das portas (lay-out). Quando as funções se tornam mais complexas o nome das entradas e saídas fornecidos pelo fabricante são escritos dentro do bloco que representa o CI, de tal modo que na parte externa possam ser usados os nomes utilizados pela aplicação. Por exemplo, o circuito integrado 7400 implementa quatro portas NAND de duas entradas e seu diagrama de pinos é dado, nesses dois formatos, confome mostrado na Figura 2.4.

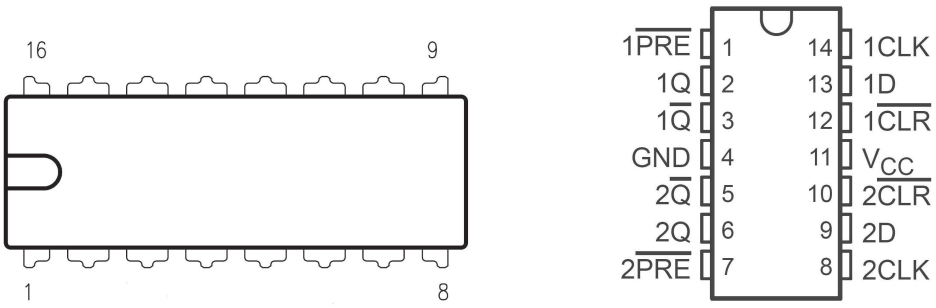


Figura 2.3 Circuito Integrado DIL da família TTL (74xx).

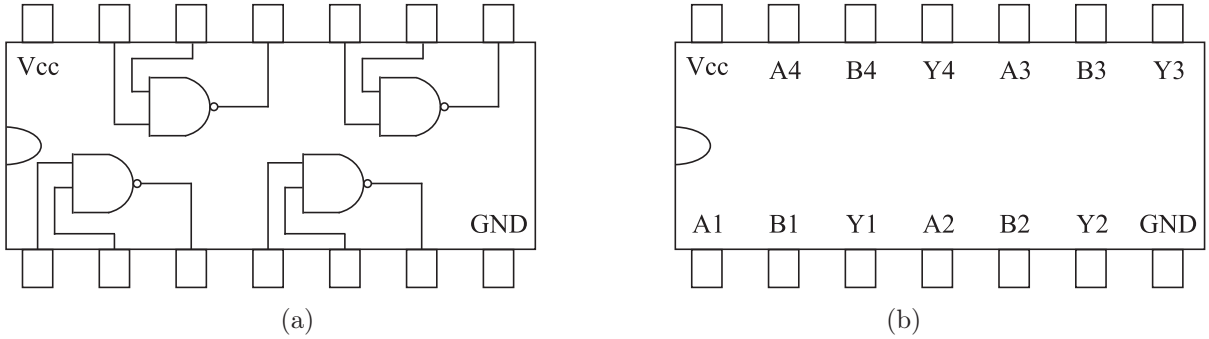


Figura 2.4 Diagramas de pinos para o 7400.

A alimentação dos circuitos TTl deve ser relaizda a partir de uma fonte de 5 Vdc, com uma tolerância de ±5 % para os da série 74. Deve ser observado que existe um número máximo de entradas que uma saída TTl de uma dada subfamília pode excitar, sem haver degradação dos níveis lógicos. Esse número máximo é denominado de *fan-out* e é necessário tomar cuidado, principalmente fquando se mistura circuitos de diferentes subfamílias. Para que o fan-out não seja excedido, é possível utilizar um ou mais buffers para aumentar a capacidade de excitação do circuito.

As condições de operação recomendadas pra a subfamília TTL padrão são resumidas na Tabela 2.5. Observar os valores correspondentes aos níveis H (2,0 V) e L(0,8 V) na entrada do circuito e a corrente máxima (16 mA) que pode ser drenada pela saída do circuito quando esta esta estiver no nível L.

Quando uma entrada TTL não está conectada a nenhum ponto elétrico conhecido, ela é dita "estar flutuando" e um nível alto é desenvolvido no terminal correspondente, ao contrário do que poderia ser suposto. Ou seja, o valor associado a uma entrada a TTL flutuando é nível H.

A família TTL fornece, com relação ao tipo de saída, três tipos de implementação de portas lógicas: saída comum, saída em coletor aberto e saída triestado.

Símbolo	Definição	Mínimo	Nominal	Máximo	Unidade
VCC	Tensão de alimentação	4,75	5	5,25	V
VIH	Tensão de entrada em nível alto	2			V
VIL	Tensão de entrada em nível baixo			0,8	V
IOH	Corrente de saída em nível alto			-400	μA
IOL	Corrente de saída em nível baixo			16	mA
TA	Temperatura de operaçõ	0		70	^{o}C

Tabela 2.5 Condições de operação recomendadas para a subfamília TTl padrão.

2.3 VERIFICANDO O FUNCIONAMENTO DE UM CIRCUITO LÓGICO

Para testar um circuito lógico, que pode ser simples como uma única porta, ou complexo como aqueles formados pela interligação de várias portas, é necessário obter a Tabela da Verdade, de modo a relacionar o comportamento da saída com as entradas aplicadas. Para tanto, é necessário ser capaz de palicar nas entradas cada uma das possíveis combinações de valores que elas podem assumir (uma para cada linha da tabela) e também ser capaz de verificar o valor da resposta.

Para gerar as variáveis lógicas de entrada podem ser usadas chaves de duas seções, uma ligada no nível alto H e outra ligada no nível baixo L. Dependendo da posição da chave, o valor gerado será H ou L. Para verificar o valor das variáveis lógicas de saída, podem ser usados indicadores luminosos comoo os Diodos Emissores de Luz (LED). Cada saída a ser verificada deve ser conectada à do circuito de polarização de um led: se o valor presente nesse terminal for H, o led estará polarizado diretamente, passará uma corrente elétrica e o led acenderá; por outro lado, se o valor for L, o led estará polarizado inversamente, não passará corrente e o led ficara apagado.

Para testar um circuito lógico é necessário, além de ligar as chave e leds, também ligar a alimentação a partir de uma fonte de tensão DC. Essa informações devem ser indicadas em um diagrama denominado de Diagrama elétrico, que mostra como os vários pinos dos CI's usados estão ligados, como também devem ser indicadas no cabeçalho da tabela da verdade usada para registrar a verificação do funcionamento. Esse diagrama elétrico deve corresponder, integralmente, ao diagrama lógico.

Para facilitar a interconexão entre terminais de saída e terminais de entrada, o circuito a ser testado é montado em uma placa especial do tipo protoboard. Essa placa de montagem é utilizada em conjunto com um Módulo de Treinamento que facilita a realização dos testes, pois fornece uma série de recursos para a aplicação de entradas e para a verificação de saídas.

2.4 CUIDADOS NA MONTAGEM E DESMONTAGEM DO CIRCUITO

- Planejamento da localização dos CI's na placa de montagem.
- Caso o circuito seja usado em outros circuitos, fazer a previsão do local de montagem de ambos.
- Na montagem deve-se seguir a disposição indicada pelo diagrama elétrico.
- Alinhar os pinos do CI, e em seguida fazer uma pressão sobre o mesmo para inseri-lo no protoboard.
- Para remover o CI utilizar um extrator de CI's ou senão uma ferramenta pontiaguda apropriada, levantando levemente uma das extremidades e em seguida a outra extremidade.
- Selecionar os fios.
- Realizar a montagem com o módulo desligado.
- Se ocorrer um comportamento anormal do circuito ao ligar o módulo, desliga-lo imediatamente.
- Ao final do experimento, guardar os CI's e os fios no local apropriado.

2.5 DEPURAÇÃO DO CIRCUITO

Caso não haja curto aparente ou o circuito não funcione com oesperado, o que é normal acontecer, ptoceder à depuração. Para tanto, usar uma ponta de prova. Na falta desta, improvisar ligando um fio comprido a um led.

Primeiramente, verificar se a montagem está de acordo com o diagrama elétrico. Observar se todos os CI's estão corretamente alimentados. Procurar por fios soltos ou por maus contatos (os fios podem estar partidos ou não estar bem encaixados).

Em seguida, a depuração pode ser realizada seguindo o fluxo de dados, ou seja, a partir das entradas. Para tanto, aplicar a entrada que produz o resultado não esperado e traçar o fluxo das entratadas, verificando o valor de cada saída intermediária, até chegar à saída. Alternativamente, essa verificação pode ser realizada a partir da saída, progredindo na direção das entradas. Em muitos casos é possível usar uma combinação dessas abordagens. de qualquer modo, o diagrama lógico deve ser usado para prever o valor esperado para cada um dos pontos testados.

2.6 MONTAGENS

2.6.1 1ª Montagem: Porta NOR de duas entradas

Verificar o funcionamento de uma porta NOR, implementada pelo CI 7402.

2.6.2 2ª Montagem: Porta AND de três entradas

Verificar o funcionamento de uma função AND de três entradas, implementada a partir de duas portas AND de duas entradas, usando o CI 7408.

2.6.3 3ª Montagem: Porta XOR/XNOR

Verificar o funcionamento do circuito lógico indicado na Figura 2.5, utilizando o CI 7486, que pode ser utilizado de três modos:

Modo 1 Portas XOR de três entradas A, B e C.

Modo 2 Porta XOR de duas entradas, com C= 0;

Modo 3 Porta XNOR de duas entradas, com C= 1;

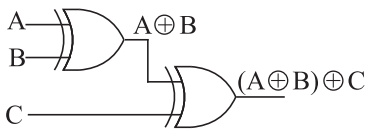


Figura 2.5 Circuito Lógico com duas portas XOR em cascata.

2.7 QUESTÕES

- Comentar a diferença entre diagrama lógico, diagrama de pinos e diagrama elétrico.
- Descrever a função obtida se no circuito da Figura 2.2(b), forem usadas duas portas NAND no lugar das portas AND.
- Como é possível obter uma função NAND de três entradas a partir de portas NAND de duas entradas?

2.8 REFERÊNCIAS BIBLIOGRÁFICAS

1. Manual de utilização e manutenção EXSTO - XD101.

2. Guia de experimentos - Lab. Circuitos Lógicos - UFCG.

3. TOCCI, Ronald Jr.; WIDNER, Neal, S.; MOSS, Gregory L. **Sistemas Digitais**. 10^{ed}. Pearson Prentice Hall, 2008. Capítulo 3.

RELATÓRIO DO EXPERIMENTO 1: PORTAS LÓGICAS

Esse é um modelo de preparação de relatório que deve ser levado devidamente preenchido para a aula de laboratório. Tal modelo se refere ao primeiro experimento. O aluno deve utiliza-lo como base para a confecção de sua própria preparação de relatório.

- Deve-se fazer um capa de identificação.
- Os objetívos são os mesmos do guia de apresentação.

3.1 MATERIAL UTILIZADO

- Módulo educacional para montagens.
- um CI 7402.
- um CI 7402.
- um CI 7486

3.2 RESUMO DA TEORIA

fazer um resumo suscinto, no máximo uma página sobre a teoria desenvolvida no experimento.

3.3 1ª MONTAGEM

3.3.1 Descrição do Funcionamento

Nesta primeira montagem verifica-se o funcionamento da porta lógica NOR, que implementa a função lógica NOR (NOT OR). Essa função é do tipo binária, pois sua descrição necessita de no mínimo duas entradas. A tabela da verdade mostrada na Tabela 3.1 descreve o funcionamento dessa função.

A expressão lógica dessa função é dada pela Equação 3.1.

$$Z = \overline{A + B} \tag{3.1}$$

O bloco lógico da função NOR é dado pelo diagrama mostrado na Figura 3.1.

A	B	$\overline{A+B}$
0	0	1
0	1	0
1	0	0
1	1	0

Tabela 3.1 Tabela da Verdade da função lógica NOR.

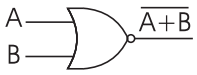


Figura 3.1 Bloco Lógico da porta NOR.

3.3.2 Diagrama elétrico

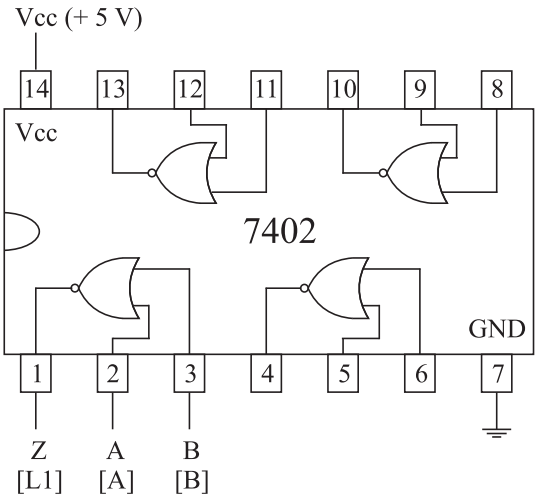


Figura 3.2 Diagrama Elétrico da 1ª montagem.

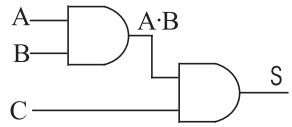
3.3.3 Verificação do Funcionamento

[A]	[B]	[L1]
A	B	Z
0	0	
0	1	
1	0	
1	1	

3.4 2ª MONTAGEM: PORTA AND DE TRÊS ENTRADAS

3.4.1 Descrição do Funcionamento

Para esta porta, o valor da saída é 1 se e somente se os valores e todas as entradas forem 1. A função lógica AND é associativa, é possível obter uma função de três variáveis a partir de duas portas AND de duas entradas, pois $S = A \cdot B \cdot C = (A \cdot B \cdot C$. Essa expressão correspode ao diagrama lógico e à tabela da verdade abaixo.



A	B	C	A·B	S
0	0	0	0	0
0	0	1	0	0
0	1	0	0	0
0	1	1	0	0
1	0	0	0	0
1	0	1	0	0
1	1	0	1	0
1	1	1	1	1

3.4.2 Diagrama elétrico

OBS. Completar o diagrama

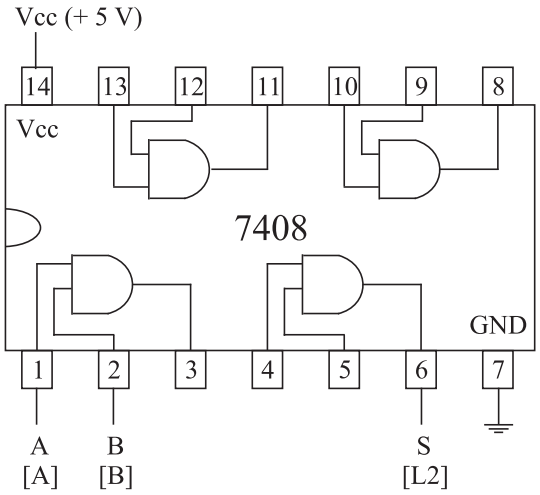


Figura 3.3 Diagrama elétrico da 2ª montagem.

3.4.3 Verificação do funcionamento

Faça um texto descrevendo do funcionamento, fazer o diagrama lógico, fazer tabela verdade do modo 3.

MODO 1: porta XOR de três entradas.

MODO 2: porta XOR de duas entradas ($C = 0$).

MODO 3: porta XNOR de duas entradas ($C = 1$)

Modo 1

A	B	C	A⊕ B	Z
0	0	0	0	0
0	0	1	0	1
0	1	0	1	1
0	1	1	1	0
1	0	0	1	1
1	0	1	1	0
1	1	0	0	0
1	1	1	0	1

Modo 2

$C = 0$			
A	B	A ⊕ B	Z
0	0	0	0
0	1	1	1
1	0	1	1
1	1	0	0

3.4.4 Diagrama Elétrico

Faça o diagrama elétrico

3.4.5 Verificação do Funcionamento

Construa as tabelas para a verificação do funcionamento

3.5 CONCLUSÕES

Escreve as conclusões sobre o experimento

3.6 QUESTÕES

Responda as questões do guia (apresente o enunciado das questões)

2º EXPERIMENTO: EQUIVALÊNCIAS DE PORTAS LÓGICAS

4.1 OBJETIVOS

- Profetar uma função lógica, como o XOR, a partir da tabela da verdade;
- usar a Tabela da Verdade para avaliar uma função lógica dada por uma expressão lógica ou por um circuito lógico.
- Usar o Teorema de Morgan para verificar a equivalência de circuitos lógicos;
- Usar a porta XOR para projetar circuitos comparadores;

4.2 INTRODUÇÃO

A função lógica binária OU-EXCLUSIVO é uma funçaõ OU que considera a exclusão mútua entre as condições relacionadas pelo conectivo lógico *ou*, de tal modo que seu valor é **1** (verdadeiro) se e somente se o valor de apenas uma de suas entradas for 1. Em outras palavras, quando os valores das entradas forem diferentes o valor da função XOR será **1**. A tabela da verdade, expressão booleana são dadas na Tabela 4.1 e o bloco funcional dessa função é dado na Figura 4.1.

A	B	$S = A \oplus B$
0	0	0
0	1	1
1	0	1
1	1	0

Tabela 4.1 Tabela da verdade.

A função XOR ou a sua negação,a função XNOR (quando os valores das entradas forem iguais o valor da função XNOR será 1), podem, portanto, ser usadas para contruir circuitos comparadores de números binários. De fato, como " dois números são iguais se

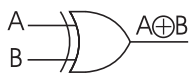


Figura 4.1 Bloco Lógico.

e somente se os bits de mesmo peso forem iguais", a função coindência (XNOR) pode ser usada para comparar cada par de bits.

Embora a função XOR esteja disponível na forma de circuito integrado, pode ser implementada a partir de funções lógicas elementares, como NEGAÇÃO, AND e OR. Para tanto, deve-se se usar a analogia com a lógica, interpretando **1** como Verdadeiro e 0 como Falso. A tabela da veridade da Figura 4.1 é, então, interpretada logicamente pela seguinte declaração:

"A saída S vale V se e somente se (a entrada A é F & a entrada B é V) ou (a entrada A é V & a entrada B é F)."

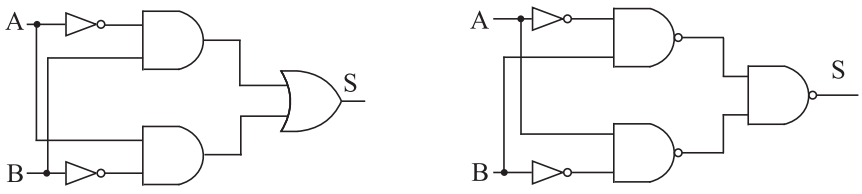
Usando a função Negação, essa declaração pode ser redefinida:

"S vale V se e somente se ($\overline{A}$ é V & B é V) ou (A é V & $\overline{B}$ é V)."

Essa declaração pode ser representada simbolicamente por umaexpressão lógica que usa o nome das variáveis e os operadores correspondentes aos conectívos lógicos presentes. Ou seja:

$$S = \overline{A} \cdot B + A \cdot \overline{B}$$

Essa expressão pode ser implementada com dois inversores, duas portas AND e uma porta OR, como indicado pelo circuito lógico da Figura 4.2(a). Esse tipo de circuito é denominado de estrutura AND-OR. COmo a estrutura AND-OR é equivalente à estrutura NAND-NAND, a função XOR também pode ser implementada apenas com portas NAND (e inversores), como indicado na Figura 4.2(b).



(a) Estutura AND-OR (b) Estutura NAND-NAND

Figura 4.2 Implementações da porta XOR.

Essa e outras equivalências entre circuitos lógicos podem ser deduzidos por meio da utilização do Teorema de De Morgan, que corresponde às seguintes identidades:

$$\overline{AB} \equiv \overline{A} + \overline{B} \quad \text{e} \quad \overline{A + B} \equiv \overline{AB}$$

A	B	AB	$\overline{\mathbf{AB}}$ ⇓	$\overline{A}$	$\overline{B}$	$\overline{\mathbf{A}} + \overline{\mathbf{B}}$ ⇓
0	0	0	1	1	1	1
0	1	0	1	1	0	1
1	0	0	1	0	1	1
1	1	1	0	0	0	0

a) $\overline{AB} \equiv \overline{A} + \overline{B}$

A	B	A+B	$\overline{\mathbf{A} + \mathbf{B}}$ ⇓	$\overline{A}$	$\overline{B}$	$\overline{\mathbf{A}} \cdot \overline{\mathbf{B}}$ ⇓
0	0	0	1	1	1	1
0	1	1	0	1	0	0
1	0	1	0	0	1	0
1	1	1	0	0	0	0

b) $\overline{A + B} \equiv \overline{A} \cdot \overline{B}$

Tabela 4.2 Verificação do Teorema de Morgan

A validade dessas identidades pode ser verificada comparando os dois lados de cada uma. Para tanto, pode ser utilizada a tabela da verdade, realizando-se a avaliação por coluna, como indicado na Tabela 4.2. Por exemplo, para a avaliação da primeira identidade, indicada na Tabela 4.2(a), em uma primeira etapa, é obtida uma coluna para AB e, em seguida uma coluna para a sua negação, que corresponde ao lado esquerdo; em uma segunda etapa, são obtidas colunas para $\overline{A}$ e $\overline{B}$, seguida de uma coluna para $\overline{A} + \overline{B}$, que corresponde ao lado direito. Comoparando essas duas colunas, como indicado por meio de setas, verificas-se que são idênticas.

De modo análogo, a segunda identidade de Morgam pode ser verificada, conforme mostrado na Tabela 4.2(b)

Sempre que for conveniente, é possível fazer uso dessas equivalências para, por exemplo, minimiza o número de portas utilizadas ou para substituir um tipo de porta lógica por outra, como indicado pelas equivalências das duas primeiras linhas da Figura 4.3.

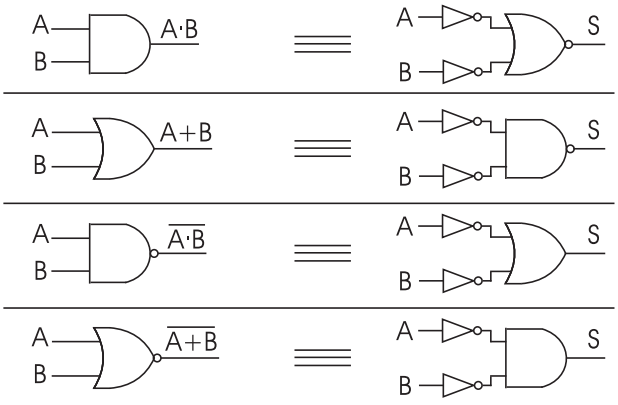


Figura 4.3 Equivalências de Morgan

As duas primeiras equivalências da Figura 4.3 são obtidas pela identidade $\overline{\overline{A}} = A$,

ou seja, *um par de negações se anula*. De fato, aplicando esse critério às expressões de Morgan:

$$AB = \overline{\overline{AB}} = \overline{\overline{A} + \overline{B}} \qquad \text{e} \qquad A + B = \overline{\overline{A + B}} = \overline{\overline{A} \cdot \overline{B}}$$

4.3 PREPARAÇÃO DO RELATÓRIO

Na seção *resumo da teoria*, que deverá ser colocada na preparação do relatório, desenvolver os seguintes tópicos:

- Descrever a função XOR e mostrar como obter sua expressão lógica;
- Dar as identidades mais usadas nas equivalências de circuitos;
- Mostrar como obter a estrutura AND-OR, graficamente, a partir da NAND-NAND;

Durante a aula, deverão ser completadas as partes relativas à verificação do funcionamento de cada montagem e as conclusões.

4.3.1 1ª Montagem: Porta XOR a partir de Porta NAND.

Verificar o funcionamento de uma porta XOR implementada com inversores e portas NAND, conforme indicado na Figura 4.2

4.3.2 2ª Montagem:Comparador de Magnitude

Projetar e verificar o funcionamento de um comparador de números binários de três bits. Usar a porta XOR para projetar um circuito capaz de comparar dois números binários A e B, de três bits cada um, conforme indicado pelo bloco funcional da Figura 4.4.

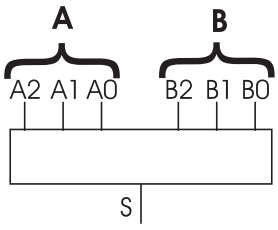


Figura 4.4 Bloco Funcional de um comparador de 3 bits

Nota: *a saída do circuito deve ser 1 somente quando $A = B$, ou seja, $A_2 = B_2$, $A_1 = B_1$ $A_0 = B_0$. Para elaborar o projeto utilizar inicialmente a porta lógica XNOR, e em seguida, por meio de equivalências de portas lógicas obter o circuito final.*

Montagem: *Observar a posição dos bits de cada número, respeitando a posição dos bits mais significatívos.*

Verificação do funcionamento: *A tabela da verdade total desse circuito possui 64 linhas, contudo para a verificação do funcionamento basta utilizar 16 linhas, 8 com resultados verdadeiros e oito com resultados falsos. Indicar o valor de A e B em decimal.*

4.3.3 3ª Montagem: Porta NAND a partir de porta OR

Verificar o funcionamento de uma porta NAND implementada com inversores e uma porta OR, como indicado na Figura 4.5.

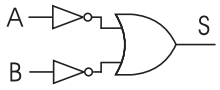


Figura 4.5 Circuito equivalente a uma porta NAND.

Nota:*Justificar essa equivalência, usando a tabela da verdade para realizar uma avaliação por coluna e usando o Teorema de Morgan.*

4.3.4 Questões

1. Mostrar a partir da tabela da verdade da porta XOR, como é possível implementar um inversor, utilizando a porta XOR.

2. Obter a função XNOR em termos de inversores e das portas AND e OR, a partir da interpretação lógica da Verdade.

3º EXPERIMENTO: SOMADOR BINÁRIO

5.1 OBJETIVOS

- Usar uma porta XOR de duas entradas para realizar a função unária de negação.
- Usar um somador binário para somar números sem sinal e com sinal.
- Verificar a ocorrência de estouro de capacidade de um somador binário, ao ser usado com os códigos mais utilizados para codificar números sem sinal ou com sinal.
- Usar um somador binário para somar e subtrair números com sinal, codificados no código 2 (Representação em Complemento de 2).
- Usar o somador implementado pelo CI 7483 como bloco construtivo de somadores com um maior número de bits.

5.2 INTRODUÇÃO

O somador implementado pelo circuito integrado 7483 pode ser usado para construir um somador binário de números com comprimento igual a quatro bits ou igual a um múltiplo de quatro. Se o comprimento for quatro bits, o CI 7483 deve ser usado da forma indicada na Figura 5.1, com a entrada de Vem-Um para o bit menos significativo (bms) igual a zero, ou seja $C_0 = 0$.

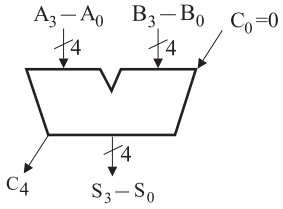


Figura 5.1 Somador Binário de quatro bits.

Para comprimentos múltiplos de quatro bits, o somador implementado pelo CI 7483 pode ser usado como um bloco funcional construtivo, utilizando o Algoritmo da Soma com Propagação do Vai-Um, para construir esse somador de maior capacidade. Assim, se o comprimento for 4xK, é necessário utilizar K blocos ligados em cascata, ou seja, a saída de Vai-Um (C_4) de cada bloco deve servir de entrada para o bloco seguinte, sendo conectada à entrada de Vem-Um (C_0).

Por outro lado, a interpretação das entradas e do resultado fornecido por um somador binário (com um número qualquer de bits) depende do código usado para a representação da informação. Por exemplo, um somador binário pode ser usado "diretamente", ou seja, sem necessidade de circuitos adicionais, para realizar operações aritméticas com números sem sinal (codificados no código 2, que usa a Representação em Complemento de 2).

Para outros tipos de códigos, é necessário incluir circuitos adicionais para corrigir o resultado. Por exemplo, o algorítmo da soma de números com sinal, codificados no Código 1 (que usa a Representação em Complemento de 1), indica que, se houver um vai-um (C_n=1), deve ser somado 1 ao resultado obtido na primeira soma para corrigí-lo, ou seja, o somador deve ser usado duas vezes.

Ao trabalhar com números com sinal, o somador binário também pode ser utilizado para realizar a subtração. Para tanto, é necessário usar um circuito adicional para realizar a operação de NEGAÇÃO. Assim, em vez de subtrai o valor do minuendo, realiza-se a soma da negação desse valor, ou seja:

$$A - B = A + (-B)$$

Ao se trabalhar com aritmética complementar, a operação de NEGAÇÃO de um número, codificado no Código 1 ou no Código 2, corresponde a, respectivamente, achar o Complemento de 1 ou de 2 desse número. Entretanto, a implementação dessa operação é simples, pois a operação de Complemento de 1 pode ser facilmente implementada por meio de inversores, enquanto que a de Complemento de 2 pode ser obtida somando-se um ao Complemento de 1.

5.3 PREPARAÇÃO DO RELATÓRIO

Na seção *Resumo da Teoria*, desenvolver os seguintes tópicos:

- Indicar com utilizar o somador binário de 4 bits (7483) para implementar um somador de oito bits (observar os nomes e valores das entradas e saídas);
- Mostrar como obter os Complementos de 1 e de 2 de um número binário B, de 4 bits.
- Mostrar como é realizada a soma para números com sinal e sem sinal, em complemento de 2.

Na parte das montagens, observar que a última montagem irá utilizar todas as anteriores. Assim, é necessário preparar cuidadosamente o Mapa de Planejamento das

Montagens.

Durante a aula, deverão ser completadas as partes relativas à verificação do Funcionamento de cada Montagem e as Conclusões.

5.4 4 MONTAGENS

5.4.1 1ª Montagem: Seletor de Função Igualdade/Complemento de 1.

(Obs.: Deixar este circuito parcialmente montado, para usar na 2ª montagem)

Verificar o funcionamento de um Seletor de Função - Igualdade/Complemento de 1 para números binários de quatro bits, dado pelo bloco funcional da Figura 5.2. A função realizada por esse circuito depende do valor da entrada de seleção SEL:

- se SEL = 0, a função selecionada é a IGUALDADE e Z = B;
- se SEL = 1, a função selecionada é o COMPLEMENTO DE 1 e Z = C_1(B).

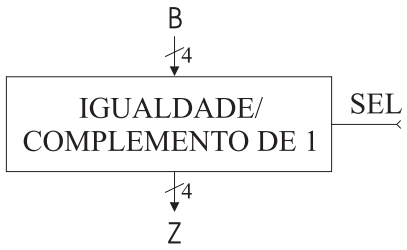


Figura 5.2 Seletor Igualdade/Complemento de 1.

a) Projetar esse circuito, considerando os seguintes pontos:

- Apesar de a Tabela da Verdade completa possuir 32 linhas, essa função pode ser descrita por uma tabela simplificada, com apenas duas linhas, usando a seguinte convenção: um valor qualquer do número B, formado pelos bits $B_3B_2B_1B_0$, pode ser representado por $b_3b_2b_1b_0$.
- A dedução lógica da expressão de cada bit B_i pode ser realizada a partir da função XOR (ver experimentos anteriores).

Observação: "Projetar" signnifica dar o Bloco Funcional, a Tabela de funcionamento, as expressões lógicas e o circuito lógico correspondente.

b) Prepara o diagrama elétrico.

c) Preparar uma tabela completa para a verificação do funcionamento desse circuito. A tabela deve ter duas colunas, uma para cada valor da entrada SEL.

5.5 2ª MONTAGEM: SOMA DE NÚMEROS SEM SINAL.

(Deixar este circuito montado, para usar na 4ª montagem)

Verificar o funcionamento do somador binário para números A e B de quatro bits, implementado pelo CI 7483. Usar o circuito seletor de função para produzir o operando B.

a) Descrever o funcionamento desse circuito, em particular a realização da soma $A + B$ (Observar o valor da entrada SEL).

b) Preparar o diagrama elétrico.

c) Preparar duas tabelas para a verificação do funcionamento desse circuito. Para tanto, faça $C_0 = 0$ e realize as somas abaixo indicadas e, em seguida, faça $C_0 = 1$ e realize as mesmas somas:

5+2	6+13	7+8	10+11	3+5	9+4	14+1	15 + 1

obs.: os números estão na base 10.

Observação: A tabela deve indicar a operação/resultado esperado (em decimal), e as entradas e saídas em binário, e o resultado obtido (em decimal) de acordo com o seguinte cabeçalho:

Operação/Resultado Esperado	$A_3A_2A_1A_0$	$B_3B_2B_1B_0$	C_0	C_4	$S_3S_2S_1S_0$	Resultado Obtido

d) Usar o Código 1 (números com sinal na representação em complemento de 1) para interpretar a operação binária correspondente à segunda linha da tabela (6+13, com C_0 igual a 0 e a 1.

5.5.1 3ª Montagem: Detetor de Estouro de Capacidade

(Deixar este circuito montado, para usar na 4ª montagem)

Verificar o funcionamento de um Detetor de Estouro de Capacidade para ser usado com um somador de números com sinal, codificados no Código 2 (representação em Complemento de 2), com quatro bits, implementado com o somador da montagem anterior.

1. Projetar o circuito que implementa essa função (usar uma saída de nome E), obtendo sua expressão a partir da interpretação lógica da tabela da verdade.

2. Preparar o diagrama elétrico.

3. Preparar a tabela da verdade para a verificação do funcionamento desse circuito.

5.5.2 4ª Montagem: Somador - Subtrrator

Verificar o funcionamento de um SOMADOR/SUBTRATOR para números com sinal, codificados no código 2 (Representação em Complemento de 2), dado pelo bloco funcional da Figura 5.3 A função realizada por este circuito depende do valor da entrada de seleção SEL: se SEL = 0, a função selecionada é a SOMA e S = A + B; se SEL =1, a função selecionada é a SUBTRAÇÃO e $S = A + C_2(B)$. A saída E indica a condição de Estouro da Capacidade.

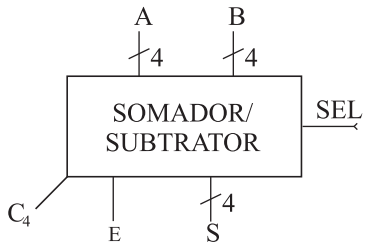


Figura 5.3 Somador/Subtrator.

a) Prejetar esse circuito usando os blocos funcionais projetados nas montagens anteriores. Para obter o Complemento de 2, indicar como usar a propriedade: $C_2(B) = C_1(B)+1$.

   I) O bloco lógico funcional desse circuito deve ser, primeiramente, detalhado na forma de um circuito lógico utilizando grandes blocos funcionais.

   II) Em seguida, esse circuito lógico deve ser detalhado usando os blocos funcionais disponíveis individualmente na forma de CI's (inversores, portas e somador).

   Obsevação: Observar, cuidadosamente, a obtenção das entradas utilizadas para o detetor de Estouro de Capacidade.

b) Preparar o diagrama elétrico completo (usar uma folha no formato paisagem).

c) Preparar duas tabelas para a verificação do funcionamento desse circuito:

   I) faça SEL = 0 e realize as seguintes somas (A +B):

   II) faça SEL = 1 e realize as seguintes subtrações (A - B):

$6+(-2)$	$7+4$	$(-3)+3$	$5+2$	$(-7)+(-4)$	$1+7$	$(-4)+(-4)$	$(-5)+(-1)$

$6-(-2)$	$7-4$	$(-3)-3$	$5-2$	$(-7)-(-4)$	$1-7$	$(-4)-(-4)$	$(-5)-(-1)$

Observação: A tabela deve indicar a operação/resultado esperado (em decimal), e as entradas e saídas em binário, e o resultado obtido (em decimal) de acordo com o seguinte cabeçalho:

Operação/Resultado Esperado	$A_3A_2A_1A_0$	$B_3B_2B_1B_0$	C_0	C_4	$S_3S_2S_1S_0$	Resultado Obtido

5.6 QUESTÕES

1. Codificar os números decimais +53 - 53 na Representação em Complemento de 2.
2. Obter os números decimais representados pelas palavras-código binárias $D5_h$, 80_h e $5A_h$, usando a Representação em Complemento de 2.
3. Qual o menor número negativo que pode ser representado com 4 e com 8 bits:
   (a) Usando a representação em complemento de 1.
   (b) Ussando a representação em complemento de 2.
4. Descrever o que é estouro de capacidade em um somador.
5. Se as entradas do 7483 forem interpretadas como números sem sinal, mostrar como pode ser detectado o estouro de Capacidade.
6. Se as entradas do 7483 forem interpretadas com números cm sinal, codificados na Representação em Complemento de 2, como é identificado o estouro de capacidade.
7. Se for usada a Representação em Complemento de 2, mostrar quantos bits devem ter um somador para realiza a operação: -32 + 11.
8. Obter um somador binário de oito bits usando o CI 7483.

4º EXPERIMENTO CIRCUITOS COMBINACIONAIS

6.1 OBJETIVOS:

- Empregar o mapa de Karnaugh para a elaboração de projetos de circuitos lógicos combinacionais.
- Projetar circuitos para a verificação de erros na transmissão de dados.
- Projetar circuitos para a comparação de magnitude entre dois números.
- Projetar circuitos combinacionais a partir de sentenças lógicas.

6.2 PREPAÇÃO DO RELATÓRIO

Na seção *Resumo da Teoria*, desenvolver os seguintes tópicos:

- Em que consiste a técnica de paridade par e impar para a verificação de erros na transmissão de dados.
- Explicar de forma sucinta as vantagens de se utilizar o mapa de Karnaugh para a elaboração de circuito lógicos.

Durante a aula, deverão ser completadas as partes relativas à verificação do Funcionamento de cada Montagem e as Conclusões.

6.3 MONTAGENS

6.3.1 1ª Montagem:Circuito gerador de paridade impar

a) Descrever o funcionamento de um circuito gerador de paridade impar de 4 bits.

b) Projetar um circuito gerador de paridade impar, para uma palavra de dados composta de 4 bits:

- Utilizar o mapa de Karnaugh.
- Apresentar tabela da verdade.
- Apresentar expressões lógicas.
- Apresentar diagrama lógico

c) Elaborar o diagrama elétrico.

d) Elaborar a tabela de verificação do funcionamento

Referências: seção 2.9 e 4.7 do livro ***Sistemas Digitais e Aplicaçõs,*** *TOCCI, 10 edição.*

6.3.2 2ª Montagem:Circuito verificador de paridade impar

a) Descrever o funcionamento de um circuito verificador de paridade impar de 4 bits.

b) Projetar um circuito verificador de paridade impar, para uma palavra de dados composta de 4 bits:

- Utilizar o mapa de Karnaugh.
- Apresentar tabela da verdade.
- Apresentar expressões lógicas.
- Apresentar diagrama lógico

c) Elaborar o diagrama elétrico.

d) Elaborar a tabela de verificação do funcionamento.

Referências: seção 2.9 e 4.7 do livro ***Sistemas Digitais e Aplicaçõs,*** *TOCCI, R. J. 10 edição.*

6.3.3 3ª Montagem: Comparador de 3 bits

a) Nessa montagem elabore um circuito que faça a comparação entre dois números X e Y, de 3 bits cada, seguindo os requisitos abaixo:

- O circuito deve possuir três saídas Z, W e U.
- Z=1 ocorre apenas se X=Y.
- W=1 ocorre apenas se X>Y.
- U=1 ocorre apenas se X<Y.

Observe que o circuito possuirá seis entradas, dessa forma, é desaconselhável o uso da tabela da verdade.

b) Elaborar o diagrama elétrico.

c) Elaborar a tabela de verificação do funcionamento.

6.3.4 4ª Montagem: Semáforo

a) Projete um sistema de sinalização de transito para o cruzamento mostrado na Figura 6.1. Sensores detectores de veículos são colocados ao longo das pistas C e D (na Rodovia) e nas pistas A e B (via de acesso). As saídas desses sensores serão nível BAIXO (0) quando nenhum veículo estiver presente e nível alto ALTO (1) quando um veículo estiver presente. O sinal de trânsito no cruzamento é controlado de acordo com a seguinte lógica:

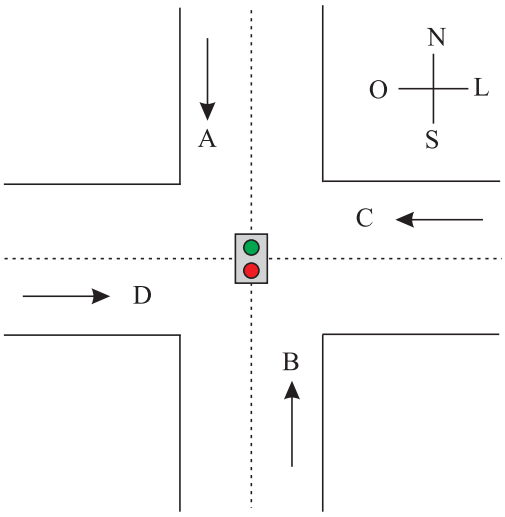


Figura 6.1 Figura da 4ª Montagem.

- O sinal da direção leste-oeste (L-O) será verde quando as duas pistas *C* e *D* estiverem ocupadas.
- O sinal da direção leste-oeste será verde sempre que as pistas C ou D estiverem ocupadas, mas com as pistas A e B desocupadas.
- O sinal da direção norte-sul (N-S) será verde sempre que as duas pistas A e B estiverem ocupadas, mas as C e D estiverem desocupadas.
- O sinal da direção norte-sul também será verde quando as pistas A ou B estiverem ocupadas enquanto ambas as pistas C e D estiverem vazias.
- O sinal da direção leste-oeste será verde quando *não* houver veículo presente.

As entradas do sistemas são os sinais dos sensores A, B, C e D, e o circuito lógico terá duas saídas NS e LO, uma para cada semáforo, cujo nível ALTO ocorrerá quando houver a luz verde.

Apresente no desenvolvimento do projeto, a tabela verdade, mapa de Karnaugh expressões lógicas e diagrama lógico.

b) Elaborar o diagrama elétrico.

c) Elaborar a tabela de verificação do funcionamento.

6.4 QUESTÕES

1. Em que caso pode se empregar técnicas de paridade para a detecção de erros na transmissão de dados de forma confiável?

2. Dado um projeto lógico descrito por uma tabela da verdade, para ele há duas opções de escolha para a elaboração: Estrutura AND-OR ou OR-AND. Por meio dos resultados da tabela da verdade, como é possível presumir qual das estruturas será a mais simples?